DTU Library

Goal-Oriented Data Collection Framework in Configuration Projects

Shafiee, Sara; Hvam, Lars; Kristjansdottir, Katrin

Published in:
Proceedings of the 2015 World Conference on Mass Customization, Personalization and Co-Creation

Publication date:
2015

Document Version
Peer reviewed version

Goal-Oriented Data Collection Framework in Configuration Projects

Sara Shafiee*, Lars Hvam, Katrin Kristjansdottir

Technical University of Denmark, Department of Engineering Management, Copenhagen, Denmark
(sashaf, lahv, katkr)@dtu.dk

Abstract. This article proposes a systematic framework for data collection when executing Product Configuration System (PCS) projects. Since the data collection in PCS is one of the most time consuming tasks, a systematic framework to handle and manage the large amount of complex data in the early stages of the PCS project is needed. The framework was developed based on the current literature in the field and revised during testing at a case company. The framework has proven to provide a structural approach for data collection, which saved the company both time and money in the initial phases of the PCS project. The framework consists of five steps, which are; establishing a goal and the methods for stakeholder analysis, categorize and group the data collection, prioritizing of products and functionalities, collection and validation of the data by domain experts and finally analysis, documentation and maintenance in the future.

Keywords: Data Collection; Product Configuration System; Stakeholders; Data Acquisition

1 Introduction

A product configurator is a subtype of software-based expert systems or Knowledge Based Systems (KBS) with a focus on the creation of product specifications [1]. Data collection in configuration projects is one of the most time consuming task due to the different expertise between domain experts and configuration engineers[1]. Therefore it

[1] Configuration Engineer models, implement, maintain PCS and also support and train users. Configuration Engineers have to create accurate product plans and manage design projects. They also are responsible for leading other staff members and keeping their knowledge of the industry as up-to-date as possible. A configuration engineer uses manual drawing tools or computer-aided design programs to create drafts of how his or her company's goods should look and operate. These drawings need to meet product specification standards and be detailed, and the products must be designed well enough to meet customer needs. Engineers in this career area also should have skills with computer programming particularly when dealing with software development, in which case their focus is on tracking and controlling changes in software products [45]

is important to scope the data collection process and use the right tools in order to reduce time and resources. Fleischanderl et al. [2] argue that up to 20% of the PCS development cost is usually spent for the configuration software system. Early in the implementation of a configuration model, a knowledge acquisition and data cleansing stage is required to centralize the product knowledge, which includes the corresponding product data [3]. Currently there is no systematic methodological framework for the knowledge acquisition processes to guide organizations to select from the appropriate application that can be used for knowledge acquisition [4]. The level of detail of the information gathered and modelled in the PCS determines the complexity of the PCS. The process of configurator's development is built around a very important element, which is to ensure that the required information is available for the project team [5].

1.1 Problem Statement

Ohno Taiichi the father of Toyota production systems and the founder of 'Just In Time (JIT)' methodology states "Making only what is needed, when it is needed, and in the amount needed!" [6]. Based on this choosing the most efficient way of collecting the right data just in the sufficient amount is necessary. Configuration engineers-spend both time and energy on gathering information, and sometimes without knowing if the gathered data are the necessary knowledge for PCS or if there is some missing datum. The main difficulties in acquiring knowledge from domain experts are due to variety of the knowledge, the various representations of the knowledge, and due to difficulties in making the knowledge explicit and accessible. Furthermore, the knowledge has to be up to date in order to secure that the PCS will provide only valid configurations. In the light of those problems for the knowledge acquisition and maintenance processes, the performance of the reasoning engine and the availability of tools for knowledge acquisition have to be taken into the account when deciding on technological basis for the configuration application [3].

This paper aims to help in the processes of controlling the exact level of detail to be included in the system, before doing the actual modeling of the PCS that result in less complexity of the PCS. In order to do so this paper defines tools to handle and manage the large amount of complex data in the early phases of the PCS project. Managing the level details in the early phases is of great importance from different aspects as it will increase the understanding, learning and make the modelling task less complex. Data collection in PCS projects is one of the most time consuming tasks due to the different expertise between domain experts and configuration engineers. There are several researches on knowledge acquisition for PCS projects but few researches have focused on the data gathering before entering and explicating them in the PCS. Therefore this paper aims to pursue that research opportunity by presenting a framework for gathering data in a more efficient way for PCS projects before they are modelled into the system.

This article's aim is to provide answers to following questions based on the framework mentioned above

1. What are the goals of the project and the stakeholders' requirements

2. How to categorize data before starting the data collection?
3. How to prioritize the products and functionalities of the PCS?
4. How to validate data?
5. How to maintain and document data?

2 Research Method

In accordance with the overall objective, the research has been structured into two phases. The first phase is focused on the development of the framework and the second phase is concerned with testing the framework.

2.1 Development of the Framework

The first phase of the research was devoted to selecting a data collection framework for IT projects from pervious literature. In this research the framework is customized based on an available framework developed for IT projects [7], methodologies and requirements for documenting PCSs [8,9], and stakeholders' analysis based on RUP principles [10].

The framework was developed by researchers with an applied research background in modelling products, product architecture, knowledge engineering and product configuration, software development, combining traditional domains of mechanical engineering with product configuration and software development.

2.2 Testing the Framework

The purpose of testing the framework in a company was to explore if the proposed framework would perform as expected. In particular, the test aimed at establishing whether the data gathered based on this framework were sufficient and efficient enough for the configuration team at the case company.

A project team was formed in an industrial company that included two researchers from the Technical University of Denmark, a software developer and configuration engineer from the company working 50% of their time on the project for three months.

3 Literature Review

The literature is identified from searching online libraries (such as Science direct, Scopus, etc.) by the use of keywords, such as "modelling techniques", "mass customization", "product configuration", "IT systems", "UML", "data gathering", "knowledge acquisition", "configuration systems structure", "knowledge management", "expert systems" and "product life cycle and data management systems". Additionally, the list of references of each article is used to identify the related bibliography, as well as the names of the researchers in the recognized research groups within this field. The first section in the literature describes the previous research works for data acquisition for IT systems in general and PCS in particular. The literature review revealed lack of structural frameworks for data gathering in PCS. The framework proposed in this article is based on the previous frameworks for data collection in IT projects, which is explained in section 3.1. In Section 3.2 previous studies and tools for stakeholder analysis are introduced, as it is fundamental in determining the goals and deliverables of the project in the early stages. There are lots of research works on efficient maintenance and documentation of the data in PCSs due to the high importance of this step which is provided in section 3.3.

3.1 Existing Frameworks for Data Acquisition for IT Projects in General and for Configuration Projects in Particular

One of the first steps in most IT projects, including PCS projects, is to collect and organize the required domain experts' knowledge. It should be noticed that a valuable source of information regarding the different aspects of products can also be available in internal software systems such as ERP system, calculation system and spreadsheet documents [11]. Felfernig et al. [12] describe how to support both goals by demonstrating the applicability of the Unified Modeling Language (UML) for configuration knowledge acquisition. Barker et al. [13] explain the volatility problems and scope expansion in the PCSs, which differentiates PCS from other IT projects. Scope expansion means that the system becomes more integrated with the other systems and as the system is used by different business groups that lead to additional requirements and data. In order to acquire knowledge in a more efficient way for the PCS with minimum time and resources consumption in the early phases of the project, more systematic methods are required. PCSs often use large and complex knowledge bases that have to be documented, maintained and updated over time. The explicit representation of problem-solving knowledge and factual knowledge can greatly enhance the role of a knowledge acquisition tool by deriving from the current knowledge base, the knowledge gaps that must be resolved [14]. Basili et al. [7] suggest a framework for collecting data in IT projects that consist of the following six schemes:

1. Establish the goals of data selection
2. Develop a list of questions of interest
3. Establish data categories
4. Design and test data collection form
5. Collect and validate data
6. Analyze the data

The configuration projects are categorized as software projects, however in configuration projects, configuration engineer utilizes information coming directly from the domain experts and internal documentation systems in order to construct the configuration model. In PCS the implementation and management are of more importance, as the obstacles are greater and data failure will have damaging consequences on the project procedure.

3.2 Stakeholder Analysis

In the knowledge acquisition process, there is a need for configuration engineer or system analyst to identify the different stakeholders and the sources of knowledge [11]. Nollore et al. [15] focus on the initial specification process in product development and propose a model to manage the interaction of the different stakeholders in the early stages. Forsythe et al. [16] define the importance of building the knowledge base and to gather data through face to face interviews between domain experts and knowledge engineers, in order to avoid practical problems in communication criteria between the knowledge engineers and the domain experts when developing the systems. Hvam et al. [17] suggest a methodology based on the representation of the product in a hierarchical structure using UML to package and present the product information for a targeted set of stakeholders (knowledge domain). In the context of configuration, at least three viewpoints are relevant: the customer, engineering and production views, which correspond to the most important stakeholders in the PCSs projects. Furthermore, Felfernig et al. [3] introduce UML as the language that is a de facto software engineering industry standard and thus more easily accessible for the stakeholders in a development project. Mortensen et al. discuss a procedure to handle the conceptual modeling, which is expected to improve the conditions to involve the relevant stakeholders early in the projects and improve conditions [18].

Based on IT projects, categorization of the requirements can be done according to two main aspects, which are functional and non-functional requirements. Non- functional requirements or general quality attributes, which emphasize that quality means compliance to requirements. A requirement that describes not what the software will do, but how the software will do it is called a nonfunctional requirement [19]. Jiao et al. [20] illustrate the steps for non-functional requirements identification, which are demonstrated in Table 1.

Table 1. Features for non-functional analysis [20]

Features	Explanation
Requirement elicitation	This is to extract and make an inventory of the requirements of stakeholders.
Requirement analysis	This is to interpret and derive explicit requirements that can be understood by everybody.
Requirement specification	Requirement specification is about the definition of concrete product specifications (FRs) in the functional domain.

A functional requirement is a requirement that specifies each function that a system must be capable of performing [19]. Lim at al. [21] provide the following features, which are demonstrated in Table 2 used to identify and prioritize those requirements.

Table 2. Features to identify and prioritize requirements [21]

Features	Explanation
Identify Requirements	The list of requirements could be elicited from interviewing an initial subset of stakeholders.
Prioritize Requirements	Prioritizes requirements using the stakeholders' ratings on the requirements and their influence in the project [21].
Recommend Requirements of Interest	Predicts a stakeholder's preference on unrated requirements using collaborative filtering techniques, and then recommends requirements with the highest predicted ratings to the stakeholder[21].
Highlight Stakeholders in Conflict	Highlights stakeholders with conflicting preferences for requirements [21].

The MoSCoW rules are commonly used when prioritizing stakeholder needs. MoSCoW is derived from the first letters of the following criteria: Must have (Mo), Should have (S), Could have (Co), Want to have (W) [22]. To improve the quality of prioritization, the analysts can merge different statements referring to the same requirement. Future work will consider crowdsourcing the stakeholders to detect duplicates and to improve the quality of the requirements, as well as integration with existing requirements management tools to support other methods of eliciting requirements (e.g., use cases, user stories, and goal modelling) [21]. Based the literature extensive researches have been done with regards to commercial IT or web-based tools to identify and categorize stakeholders. In Table 3 the current literature for stakeholder analysis, both for the IT projects and PCS projects, is summarized.

Table 3. Stakeholders analysis literature

References for Stakeholders' analyses	IT Projects	PCS Projects
Forsythe et al. (1989) [16]		✓
Ebert et al. (1997) [19]	✓	
Jiao et al. (2006) [20]	✓	
Hvam et al. (2008)[17]		✓
Lim at al. (2011) [21]	✓	
Nollore et al. (1999) [15]		✓
Felfernig et al. (2014) [3]		✓
Bittner (2002) [22]	✓	
Mortensen et al. (2008) [18]		✓

3.3 Validation, Test, Maintenance and Documentation of the Data

In industrial companies delivering complex and highly engineered products, it is crucial to have an efficient system for the documentation of attributes and rules implemented in the PCS, which enables communication of the product knowledge with domain experts. The documentation is also important for the configuration engineering team working with the PCSs to enable them to do future development and maintenance of the systems. Studies in companies using product PCSs have revealed that without documentation system companies are unable to develop their configurators, which can lead that they are forced to abandon or rebuild PCS [23]. It is therefore of importance to have reliable product documentation, i.e. without technical errors and mirroring exactly the customer's expectations [24]. Documentation is vital for all IT projects as it is used for sharing knowledge between people and it reduces knowledge loss when team members become unavailable [25].The underlying product model is a 'living organism' and will quickly become obsolete if not maintained [26]. Tiihonen et al. [27] reflect on the challenges of using PCSs and one of them is that many practical configuration models are poorly documented, incomplete, difficult to understand or outdated. The maintenance of the PCSs can be divided into two general areas: maintenance of the product model and maintenance of the IT system [26].

4 Framework Development

To avoid the risk of failure when using data collection methods, this framework helps configuration engineers and the organizations to become more efficient in this processes. Timeliness of data collection and data validation is quite important for the accuracy of the development. In the area of PCS, the knowledge to be acquired can be both unstable and contentiously changing [11]. As the system grows and get more successful, the users expect more and have new requirements [13]. That is why an iterative framework is needed so it can be used during the project development as well as after the development and in the production phase. Based on Basili's [7] six steps approach, which was explained previously in the literature, it is possible to specify the data collection framework for a configuration project listed below:

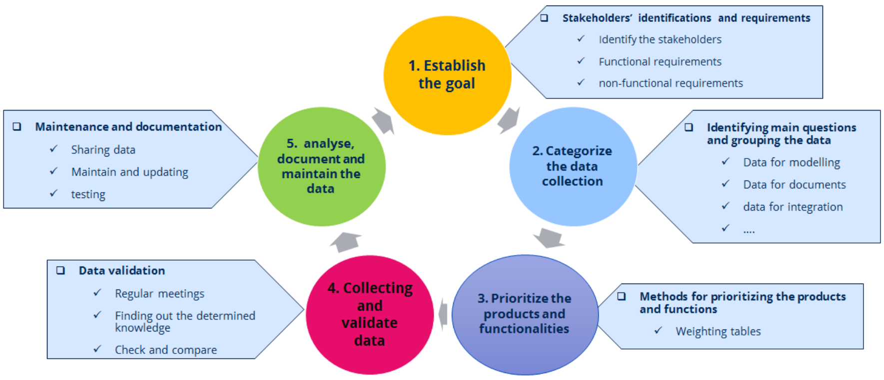

Fig. 1. The proposed data collection framework illustration for configuration projects

The suggested framework is built on a five steps that can be used iteratively. In order to accomplish the framework all the sub steps listed under each step have to be finished. Aligned with different projects in different companies and with different types of stakeholders the sub steps might have to be adjusted. This framework should enable the configuration engineers to be more in control of the level of details to be included in the project by knowing the exact outputs and thereby being able to ask for the relevant input needed for the development of the PCS.

4.1 Establish the Goal of the Data Collection

Considering product configuration as a requirement for highly engineered products, the team needs to understand what kinds of outputs are needed for the project accomplishment. A goal determination is used to increase the understanding of the environment and to focus the attention on techniques that are useful at this stage [7]. Jiao et al. [20] defines the customer's requirements in general in three aspects which are; requirements elicitation, requirements analysis, and requirements specification. Based on the RUP methods, the stakeholders and their necessities can be drawn through two specific methods: the first one is by using process flowcharts (TO-BE process) [17], and the second one is by utilizing the use case diagrams from the RUP method [10]. The TO-BE flowcharts can be drawn according to different scenarios to determine the future process [28]. A use case is a pattern for a limited interaction between a system and actors in the area of application. Use case diagrams are the means of expressing the requirements and the actors involved in the project. According to the RUP rules the same use case is utilized in system analysis, design, implementation and testing [17].

4.2 Categorize the Data Collection

The most efficient way of data categorization is to determine the most important output data according to the stakeholders' requirements and subcategorizes step by step. Such grouping permits the type of data in the PCS project with respect to the needed data. In the configuration projects, the data needed can directly come from the stakeholders' requirements determination e.g. data for documents needed from configurator, data needed for integration and software development, data needed for price calculations and etc. This way the data needed can be categorized according to the desired deliverables.

4.3 Prioritizing the Products and Functionalities

Component-based development concerned with how to build quality systems that satisfies the business needs quickly, preferably by using parts rather than handcrafting every individual element [29]. The purpose of using a component based structure based on RUP approach [10] is to break a large and complicated project into smaller pieces in order to make the process easier both for the users and developers. As Felfering et al. [3] describe using a component based strategy can be very helpful in complicated and highly engineered projects. When categorizing the expected results and outputs from the configurator due to the requests from stakeholders, the expectations from the project become clearer. By giving components weight to determine their priorities and importance in the project can help in the initial assessment [30]. The weight can be an index for scoring the value adding activities. The comparison between the tables related to the components weights gives a sense of the importance of the components regarding different aspects such as stakeholders' requirements, sales rates, market needs or even the complexity of the component.

4.4 Collecting and Validating the Data

The accuracy and correctness of data is checked by the domain expert for correctness, consistency, and completeness during the project iteration [30] in each version of PCS. A number of methods have been used to help the engineers to do project tests iteratively [31]. In order to gather the information the best option is often to have regular meetings to ask for the knowledge, receive feedback and validation.

4.5 Analyzing, Maintaining and Documenting the Data

Documentation systems are one of the vital tools during the project development, which permit the domain experts to be involved in the process from the first phases of the PCS project. The presentation of the knowledge in the PCS projects in the phenomenon model structure is one of the greatest challenges in these kinds of projects [32,33]. The ideal situation is to have a documentation system and exchange the knowledge inside the PCS with domain experts to allow them to test, verify and update the knowledge in the system iteratively. This can reduce costs due to preventing potential mistakes in the final stages of the project. The results indicate that having documentation system available during the system development reduces the maintenance time by approximately 20 percent [34]. There are a couple of authors had been worked on representation techniques [8,35,36,37,38,39,40]. In this step it is recommended to use Product Variant Master (PVM) associated with Class, Responsibilities, and Collaboration (CRC) Cards and class diagrams, which are built on UML. The reason for the selection of this representation technique is based on experience of the research team.

Product Variant Master. The PVM presented by Hvam [41] represents the product knowledge in a structured format from three different aspects, which are customer's view, engineering view and production/part view. The different aspects are chosen to represent the most important stakeholders of the system. Furthermore, the relation between the different views allows identification of none value adding activates and complexity. The PVM is built of the Product Family Master Plan that is used developing "product families", based on the architecture presented by Harlou [42]. For visualizing and facilitating product knowledge, the PVM has proven to be successful in numbers of cases.

CRC Cards. The CRC cards were first proposed by Cunningham [43] as a way to teach object oriented thinking. Hvam et al [41] later presented several revised definitions of the CRC cards to be used in product configuration projects. The CRC cards are used associated with the PVM and the class diagram in order to contain more detailed information.

5 Case Study

The proposed framework was tested at an industrial engineering company, which is specialized in production of heterogeneous catalysts and in the design of process plants based on catalytic processes. The framework was used in the early phases in PCS project for Wet Sulphur Acid (WSA) processes plants used in industries like oil refining, coking, coal gasification and viscose fiber use.

5.1 Establish the Goal of the Data Collection:

Aligned with the literature, workshops where held in order to determine the goal of the data collection after determining the main stakeholders. The main tools that were used in this phase were flowcharts to determine the To-Be processes and use case diagrams for visualization and communication with the domain experts. A long list of functional and nonfunctional requirements for individuals parts of the system where identified. In Table 4 some of the stakeholders' requirements have been prioritized according to the MoSCoW principles.

Table 4. Examples of stakeholders' requirement prioritization

List of requests	Must have	Should have	Could have	Want to have
Combining document snippets into full technical or commercial proposals (sales people and cost estimators)		✓		
Loading data from the configurator into tables in the technical and commercial (sales, cost estimators and marketing group)			✓	
Price calculation, bills of material and scope of supply (all stakeholders)	✓			
Having colors for different components in user interface				✓

In this case use case diagrams used for the project visualizing where the main actors involved in the configuration processes along with the functional requirements.

5.2 Categorize the Data Collection

At this stage of the project, there is a need to determine what kinds of information are needed based on the stakeholders' requirements and the project goal. In Table 5 a categorization of the information is listed.

Table 5. The categorized phases for the case study

Categorized phase	Output needed
Configuration requirements	There is a need for the products data for configuring the product according to the stakeholders' order in the execution of the system.
Calculation pre-requirements	There is a need for the data are used in the calculation inside the configuration engine for constraint parts.
Needed documentation requirements	There is a need for the data are used in the documentation part for Price Calculation Sheets (PCS), Bills of Materials (BOM), Scope of Supply (SOP) and….
Integration requirements	There is a need for data is used for the integration section. ✓ For calculation ✓ For flow diagrams

5.3 Prioritizing the Products and Data

Weighting tables are used to determine the importance of different components [30]. In this case, the weighting is according to the stakeholders' requirements and the complexity of the components. It means the project will start with the components that receive the highest score according to the stakeholders' needs and the lowest score for the complexity. The complexity in this case is determined by comparing number of rules and attributes across different products. Products categorized with low complexity therefore have few rules and attributes compared to other products. In Table 6 an example of component weighting is shown. In this table the product is requested highly by the stakeholders but also earns high degree of complexity. The weighting table could contain other factors to make the decision making easier for the configuration group.

Table 6. Weighting Tables

Product 1	**Importance (0-10)**	**Complexity (0-10)**
Stakeholder 1	10	10
Stakeholder 2	9	6
Stakeholder 3	8	9
Stakeholder 4	10	10
Stakeholder 5	8	8
Stakeholder n	….	….
Mean value of importance	$\frac{10+9+8+10+8+\cdots}{\text{n}}$	$\frac{10+6+9+10+8+\cdots}{\text{n}}$

5.4 Collection and Validation of the Data

In this particular case, the close relation with domain experts was really helpful to gather and validate data for the project. In this step the following achievements are fundamental for the project success:

- Logical consistency: the attributes, variables and constraints should be consistence when entering the PCS.
- Validate the model with domain experts: there must be an efficient communication method available between the configuration group and domain experts. Therefore, domain experts are able to check and validate all the knowledge modeled in the PCS. A communication system based on the PCS data extraction used in this case [9].

5.5 Analyzing, Maintaining and Documenting the Data

The documentation system at the company illustrates the knowledge in the PCS in the form of PVM and class diagrams. The system has been developed to have a proper communication with the domain expert during the project development as well as for the documentation and maintenance of the knowledge and for the future updates and changes. An example of the PVM that was made for this project is shown in Figure 2.

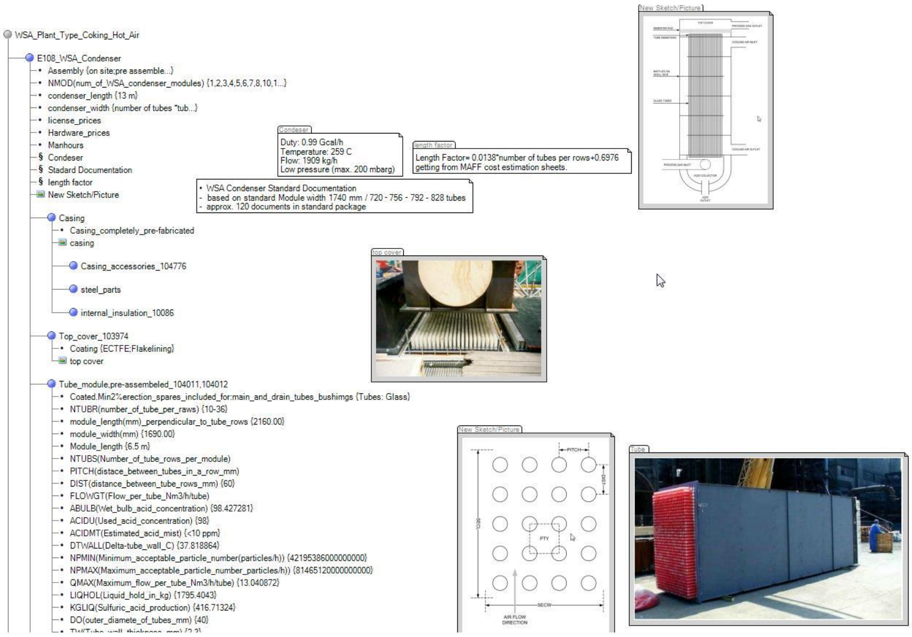

Fig. 2. The example for PVM from the case study

6 Discussions and Conclusion

The suggested framework for data collection is developed based on literature and experiences from implementing PCSs and IT projects. Companies investing in PCSs aim to use the PCS as a solution for decreasing complexity; and make the sales and engineering processes more efficient [44]. Without a clear framework for data gathering from the early stages, the PCS tends to get complicated as a result to lack of focus on the level of the data details. The framework proved to be useful for the project team by supporting early clarification of the project goal, identification of stakeholders' requirements, data categorization, products prioritization and finally for the validation of the data and maintenance and documentation. The framework helped to focus and give priority only to needed parts of the PCSs and reduce time spent in the early phase of the project. The suggested framework has been tested in an ETO company on a couple of PCS projects. In terms of future studies several areas have been identified that are listed below:

- More testing for different types of project and in different companies.
- More research on the categorization of data
- Other available tools and methods for prioritization of the products and functionalities

Acknowledgement

This work was conducted within a Company called Haldor Topsoe A/S. The granted project and financial support is gratefully acknowledged.

References

1. Haug, A., Hvam, L., Mortensen, N. H.: Definition and Evaluation of Product Configurators Development Strategies. In: Computers in Industry. 63, 471-481(2012)
2. Fleischanderl, G., Friedrich, G. E., Haselböck, A., Schreiner, H., Stumptner, M.: Configuring large systems using generative constraint satisfaction. In: IEEE Intelligent Systems. 13, 59-68 (1998)
3. Felfernig, A., Hotz, L., Bagley, C., Tiihonen, J.: Knowledge-based Configuration: From Research to Business Cases. Morgan Kaufman (2014)
4. Mildred, L., Shaw, G., Woodward, J.: Modeling expert knowledge. In: Knowledge Acquisition. 2, 179-206 (1990)
5. Turley, F.: An Introduction to PRINCE2®. Management Plaza (2010)
6. Taiichi, O.: Toyota Production System: Beyond Large-Scale Production (English translation ed.). Oregon: Productivity Press , Portland (1988)
7. Basili, V. R., Weiss, D. M.: A Methodology for Collecting Valid Software Engineering Data. In: IEE transactions o software engineering.10, 728-738 (1984)
8. Haug, A.: Representation of Industrial Knowledge – As a Basis for Developing and Maintaining Product Configurators. PhD thesis: Lyngby, Technical University of Denmark (2007)
9. Shafiee, S., Hvam, L., Kristjansdottir, K.: An agile documentation system for highly engineered, complex product configuration systems. In: 22nd EurOMA Conference: Operations Management in an Innovation Economy Neuchatel, Switzerland (2015)
10. Kruchten, P.: The Rational Unified Process: An Introduction. Addison-Wesley, New York (1998)
11. Friedrich, G., Jannach D., Stumptner, M., Zanker, M.: Knowledge Engineering for Configuration Systems. In: Knowledge-based configuration: From research to business cases, pp. 139-155, Morgan Kaufmann (2014)
12. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M., Zanker, M.: UML as knowledge acquisition front end for Semantic Web configuration knowledge bases. In: CEUR Workshop Proceedings (2002)
13. Barker, V. E., O'Connor, D. E., Bachant, J., Soloway, E.: Expert systems for configuration at Digital: XCON and beyond. Communications of the ACM. 32, 298-318 (1989)
14. Ramachandran, S., Gil, Y.: Knowledge acquisition for configuration tasks: The EXPECT approach. In: AAAI Workshop on Configuration (1999)
15. Nellore, R., Söderquist, K., Eriksson, K. Å.: A specification model for product development. European Management Journal.17, 50-63 (1999)
16. Forsythe, D. E., Buchanan, B. G.: Knowledge acquisition for expert systems: Some pitfalls and suggestions. IEEE Systems, Man, and Cybernetics Society.19, 435 – 442 (1989)
17. Hvam, L., Mortensen, N. H., Riis, J.: Product Customization, Springer, Berlin (2008)
18. Mortensen, N. H., Harlou, U., Haug, A.: Improving decision making in the early phases of configuration projects. International Journal of Industrial Engineering: Theory, Applications and Practice.15, 185-194 (2008)
19. Ebert, C.: Dealing with nonfunctional krav in large software systems. Journal of Software Engineering. 3, 367-395 (1997)

20. Jiao, J., Chen, C.: Customer requirement management in product development: a review of research issues. Concurrent Engineering. 14, 173-185 (2006)
21. Lim, S. L., Damian, D., Finkelstein, A.: StakeSource2. 0: using social networks of stakeholders to identify and prioritise requirements. In: Proceedings of the 33rd international conference on Software engineering. ACM (2011)
22. Bittner, K.: Use case modeling. Addison-Wesley Longman Publishing Co., Inc. (2002)
23. Haug, A., Hvam, L., Mortensen, N. H.: Implementation of conceptual product models into configurators: From months to minutes. In: 5th World Conference on Mass Customization & Personalization MCP (2009)
24. Forza, C., Salvador, F.: Managing for Variety in the Order Acquisition and Fulfillment Process: The Contribution of Product Configuration Systems. International Journal of Production Economics. 76, 87-98 (2002)
25. Paetsch, F., Eberlein, A., Maurer, F.: Requirements engineering and agile software development. IEEE (2003)
26. Hvam, L., Pape, S., Nielsen, M. K:. Improving the quotation process with product configuration. Computers in Induustry, 607-621 (2006)
27. Tiihonen, J., Heiskala, M., Anderson, A., Soininen T.: WeCoTin–A practical logic-based sales configurator. AI Communications. 26, 99-131 (2013)
28. Hvam, L.: Mass customisation in the electronics industry: based on modular products and product configuration. International Journal of Mass Customization. 1, 410-426 (2006)
29. Tiihonen, J., Soininen, T., Männistö, T., Sulonen, R.: State of the practice in product configuration–a survey of 10 cases in the finnish industry. Knowledge intensive CAD, 95-114 (1996)
30. Shafiee, S., Hvam, L., Bonev, M.: Scoping a Product Configuration Project for Engineer-to-Order. International Journal of Industrial Engineering and Management (IJIEM).5, 207-220 (2014)
31. Hirsch, M.: Making RUP Agile. OOPSLA 2002 Practitioner Report (2002)
32. Sabin, D., Weigel, R.: Product Configuration Framework-a survey. IEEE Intelligent Systems and their Applications.13, 42-49 (1998)
33. Hansen, B.: Development of Industrial Variant Specification Systems. PhD thesis. Department of Manufacturing Engineering and Management, Technical University of Denmark, Lyngby, Denmark (2003)
34. Tryggeseth, E.: Report from an Experiment: Impact of Documentation on Maintenance. Empirical Software Engineering: An International Journal. 2, 201-207 (1997)
35. Aldanondo, M., Rouge, S., Veron, M.: Expert Configurator for Concurrent Engineering: Cameleon Software and Model. Journal of Intelligent Manufacturing. 11, 127-134 (2000)
36. Chao, P., Chen, T.: Analysis if Assembly Through Product Configuration. In: Computers in Industry. 44, 189-203 (2001)
37. Margo, D., Torasso, P.: Decomposition Strategies for Configuration Problems. In: Artificial Intelligence for Engineering Design Analysis and Manufacturing.17, 51-73 (2003)
38. Tseng, H., Chang, C., Chang, S.: Applying Case-Based Reasoning for Product Configuration in Mass Customization Environment. Expert Systems with Applications.29, 913-925 (2005)
39. Jinsong, Z., Qifu, W., Li, W., Yifang, Z.: Configuration Oriented Product Modelling and Knowledge Management for Made-to-order Manufacturing Enterprises. International Journal of Advanced Manufacturing Technoligy. 25, 41-52 (2005)
40. Yang, D., Miao, R., Wu, H., Zhou, Z.: Product Configuration Knowledge Modelling Using Ontology Web Language. Expert System with Applications. 36, 4399-4411 (2009)

41. Hvam, L.: A procedure for the application of product modelling. International Journal of Production Research. 39,873-885 (2001)
42. Harlou, D.: Developing Product Families Based on Architectures Contribution to a Theory of Product Families. Technical University of Denmark, Lyngby (2006)
43. Beck, K., Cunningham, W.: A Laboratory for Teaching Object-Oriented Thinking, Object-Oriented Programming Systems. Languages and Applications, 1-6 (1989)
44. Hippel, E.: User toolkits for innovation. Journal of product innovation management.18, 247-257 (2001)
45. "WiseGEEK,http://www.wisegeek.com/what-does-a-configuration-engineer-do.htm